



Introdução IA em Python

Bem-vindo ao nosso estudo sobre a Introdução em IA em Python. Este estudo ajudará você a desenvolver programas na linguagem Python para resolver problemas com ênfase em Inteligência Artificial. Bons estudos!

Algoritmos em IA

Existem vários algoritmos para resolver problemas em Inteligência Artificial. A linguagem Python é a linguagem preferencial para desenvolver esses algoritmos porque tem pacotes prontos que contêm algoritmos já implementados para resolver muitos problemas de Inteligência Artificial.

Algoritmo KNN

Um dos algoritmos mais simples de implementar de Machine Learning é o algoritmo KNN. O algoritmo KNN, que significa K-nearest neighbors - K-vizinhos mais próximos - está entre os mais simples de todos os algoritmos e um dos primeiros a serem aprendidos por quem estiver iniciando os seus estudos em aprendizado de máquina. Esse algoritmo é utilizado para classificação e para regressão, podendo ser usado para atribuir peso às contribuições dos vizinhos, onde a média é calculada com peso maior para os vizinhos mais próximos do que para os mais distantes.

Resumindo, de acordo com a amostra de um conjunto de dados o algoritmo avalia a sua distância em relação aos vizinhos mais próximos. Se entre os vizinhos mais próximos tiver uma maioria de determinada classe, a amostra será classificada nesta categoria. Nessa aula vamos entender melhor como funciona esse algoritmo KNN e vamos fazer a implementação de um código utilizando essa técnica de classificação.

Exemplo prático de implementação em Python

Vamos fazer um exemplo de implementação em Python. Na linguagem Python, vamos utilizar o pacote scikit-Learn, que possui o KNN implementado. A única coisa que precisamos fazer é passar os parâmetros da forma certa. Como vamos utilizar esse pacote, precisamos importá-lo porque ele não vem com a instalação padrão do Python, então o ideal é utilizar o Jupyter Notebook, que já vem com os pacotes importantes para ciência de dados. Se não quiser instalar o Anaconda para usar o Jupyter Notebook, podemos utilizá-lo de modo online pelo Google Colab.

O algoritmo precisa de uma base de dados com exemplos para poder calcular as distâncias e prever a qual classe pertence o novo dado. No nosso exemplo vamos usar um dataset experimental (dados) que classifica tipos de livros, para saber se o livro é didático ou não.

Para saber se o livro é didático, vamos reunir algumas características de livros:

O livro é de fácil leitura? (nota de 0 a 5)

O livro é difícil? (nota de 0 a 5)

O livro é cheio de fórmulas? (nota de 0 a 5)

O livro contém rimas? (nota de 0 a 5)

O livro é utópico? (nota de 0 a 5)

Então temos o dataset (base de dados experimental):

```
livro1= [0,5,5,0,0]
livro2= [0,5,0,0,0]
livro3= [5,5,0,0,0]
livro4=[0,0,5,5,0]

isDidatico = [0,1,1,0]
```

A variável isDidatico contém uma lista contendo as classes que são atribuídas a esse tipo de livro.

Veja a implementação no Google Colab:



```
# livros [ didático, poesia, ficção, auto-ajuda, economia]
'''
O livro é de fácil leitura? (nota de 0 a 5)
O livro é difícil? (nota de 0 a 5)
O livro é cheio de fórmulas? (nota de 0 a 5)
O livro contém rimas? (nota de 0 a 5)
O livro é utópico? (nota de 0 a 5)
'''

livro1= [0,5,5,0,0]
livro2= [0,5,0,0,0]
livro3= [5,5,0,0,0]
livro4= [0,0,5,5,0]
isDidatico = [0,1,1,0]  # as classes que são atribuídas ao livro didático

X = [livro1,livro2,livro3,livro4]
y = isDidatico

from sklearn.neighbors import KNeighborsClassifier

neigh = KNeighborsClassifier(n_neighbors=3)

neigh.fit(X, y)

novoLivro = [[0,2,4,0,0]]
result = neigh.predict(novoLivro)
print(result)

if result == 0 :
    print("É didático")
else:
    print("não é didático")
```

```
[0]
É didático
```

Neste exemplo o resultado mostra que o livro é didático, pois a distância do novo livro é menor para o(s) livro(s) didático(s).

Resumo do processo que realiza o algoritmo KNN:

1) Recebe um dado não classificado e mede a distância do novo dado em relação a cada um dos outros dados que já estão classificados;

2) Seleciona as K menores distâncias;

3) Verifica as classes dos dados que tiveram as K menores distâncias e conta quantas vezes cada classe apareceu;

4) O novo dado é classificado como pertencente à classe que mais apareceu.

Atividade Extra

Para se aprofundar no assunto desta aula, pesquise mais sobre o algoritmo KNN e implemente outros exemplos em Python. Uma boa opção para essa pesquisa é o livro Data Science do Zero, de Joel Grus.

Referência Bibliográfica

GRUS, Joel. **Data Science do Zero:** Primeiras regras com o Python. Rio de Janeiro: Alta Books, 2016.

IZBICKI, Rafael, SANTOS, Tiago M. Aprendizado de Máquina: **Uma abordagem estatística**. São Carlos: Rafael Izbicki, 2020.

Ir para exercício